

Student Name:

Student id:

Sect #:

Serial:

University of Bahrain

College of Information Technology
Department of Computer Science

ITCS242: Assembly Language Programming

Quiz #2: Fundamentals

- Answer the next 3 questions as needed

- 1) Give ONE instruction that uses indirect operand: `MOV ECX, [ESI]`
- 2) The range of values that can be stored using `sbyte` is `80 H` and `7F H`
- 3) The directive that defines a constant `UU` equals to “ITCS242!” is `UU EQU “ITCS242!”`.

- Consider the following directive and answer the next 3 questions.

```
UU  sword  2A5CH,8FA4H,2 dup (-20,3 dup (-5, 2AH),7F4AH,9C4H)
```

- 4) The instruction that stores in `AX` register the number of bytes of array `UU` is

```
MOV  AX,  SIZEOF  UU
```

- 5) The instruction that swaps the first word of `UU` with `BX` is

```
XCHG  BX,  UU.
```

- 6) The instruction that stores the value of `CX` in the **last** word of array `UU` in the above directive is

```
MOV  UU[sizeof UU - 2], CX
```

- Given an array: `ME SDWORD 240 dup (?)`; Write Assembly code that swaps the words in each element of array `ME`.

```
MOV  ECX, LENGTHOF ME
MOV  EBX, 0
L1:  MOV  AX, WORD PTR ME[4*EBX]
      XCHG AX, WORD PTR ME[4*EBX+2]
      MOV  WORD PTR ME[4*EBX], AX
      INC  EBX
      LOOP L1
```

Student Name:

Student id:

Sect #:

Serial:

University of Bahrain

College of Information Technology
Department of Computer Science

ITCS242: Assembly Language Programming

Quiz #2: Fundamentals

- Answer the next 3 questions as needed

- 1) The range of values that can be stored using **word** is **8000** H and **7FFF** H
- 2) Give ONE instruction that uses indexed operand: **DEC TT[EBX]** .
- 3) Give ONE instruction that exercises the size ambiguity problem: **INC [EBX]** .

- Consider the following directive and answer the next 3 questions.

TT word 2A75H, 9F64H, A6F9H, 2 dup(4, 3 dup(-40, 2AH), 7F02H)

- 1) The instruction that stores in **AX** register the number of items in array **TT** is

MOV AX, LENGTHOF TT

- 2) The instruction that makes **ebx** point to the last word of array **TT** is

LEA ebx, TT[sizeof TT-2]

- 3) The instruction that extends the first word of array **TT** into **EBX** register is

MOVSX EBX, TT

- Given an array: **YOU WORD 120 dup (?)**; Write Assembly code that stores +1 in the first byte and -1 in the second byte in each word of array **YOU**.

MOV ECX, LENGTHOF YOU

MOV EBX, 0

L2: MOV YOU[2*EBX], 01FFH

INC EBX

LOOP L2